UNITED STATES PATENT AND TRADEMARK OFFICE

Appl. no. 09/954,596	Filed: September 12, 2001
Examiner: ELLIS, RICHARD L	Art unit: 2183
Confirmation no. 8857	Reference: FREIP033-2

CORRECTED BRIEF ON APPEAL

This is an appeal of a final rejection dated March 24, 2004. Appeal was timely taken on August 26, 2004. Previously filed are Forms PTO/SB/22, PTO/SB/17, and PTO-2038 in payment of the fee set forth in 37 CFR § 41.20(b)(2). This corrected brief is submitted in response to the Examiner's Notice of Defective Appeal Brief dated March 21, 2005.

Real party in interest[1]

The real party in interest is Anton Gunzinger of Zürich, Switzerland.

Related appeals and interferences

Applicant is unaware of any application, patent, appeal or interference or any other prior or pending appeals, interferences or judicial proceedings which may be related to, directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

Status of claims

Claims 1-15 and 32 are canceled. Claims 16-31 and 33-34 are rejected. The rejection of claims 16-31 and 33-34 is appealed.

1 The Examiner has asked that the headings pursuant to 37 CFR section 41.37(c)(1) be rearranged so that each is above the respective item rather than in the same line with the respective item. For that reason, this heading and all subsequent headings have been rearranged as requested.

Status of amendments

An amendment after final rejection was filed August 17, 2004. By a paper dated September 21, 2004, the Examiner refused to enter the amendment.

Summary of claimed subject matter

A concise explanation of the subject matter defined in each of the independent claims involved in the appeal, which shall refer to the specification by page and line number, and to the drawing, if any, by reference characters.

Claim 16. A method of operating a parallel computer system having at least first and second processor elements (page 5, line 19; Fig. 2, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:

writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of

the first processor element (page 6, lines 3-5);

adding, by the communications manager, a global address to the result of the computation (page 6, line 5);

propagating, on the message-passing communications network, a message comprising the global address and the result of the computation (page 6, lines 5-9);

receiving the message, via the message-passing communications network, by the communications manager of the second processor element (page 6, lines 8-9);

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match (page 6, lines 9-14);

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17).

Claim 21. A parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the local data memories of the at least first and second processor elements not on a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor elements each executing an application (page 5, line 29); each communications manager

further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means (page 6, line 5) responsive to writing (page 6, lines 3-5), by the processor of the processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating (page 6, lines 5-9), on the message-passing communications network, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means (page 6, lines 8-9) responsive to receiving a message, via the message-passing communications network, by the communications manager, for comparing (page 6, lines 9-14) the global address in the message with the plurality of predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17).

Claim 26. A method of operating a parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor

elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:

writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element (page 6, lines 3-5);

adding, by the communications manager, a global address to the result of the computation (page 6, line 5);

propagating, on the message-passing communications network, a message comprising the global address and the result of the computation (page 6, lines 5-9);

receiving the message, via the message-passing communications network, by the communications manager of the second processor element (page 6, lines 8-9);

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match (page 6, lines 9-14);

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17),

wherein the predefined values are further characterized as comprising an address window, each window comprising an initial address and an end address, a match comprising the global address falling between the initial address and the end address (page 6, line 31 to page 7, line 5; Fig. 3, reference characters 201' and 302').

Claim 27. A method of operating a parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:

writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element (page 6, lines 3-5);

adding, by the communications manager, a global address to the result of the computation (page 6, line 5);

propagating, on the message-passing communications network, a message comprising the global address and the result of the computation (page 6, lines 5-9);

receiving the message, via the message-passing communications network, by the communications manager of the second processor element (page 6, lines 8-9);

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match (page 6, lines 9-14);

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17),

wherein the propagating step comprises propagating the message to a number of processor elements, the number comprising less than all and more than one of the processor elements (Fig. 3, reference character 311 intersecting less than all and more than one of the processing elements).

Claim 28. A parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the local data memories of the at least first and second processor elements not on a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means (page 6, line 5) responsive to writing (page 6, lines 3-5), by the processor of the processor

element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating (page 6, lines 5-9), on the message-passing communications network, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means (page 6, lines 8-9) responsive to receiving a message, via the message-passing communications network, by the communications manager, for comparing (page 6, lines 9-14) the global address in the message with the predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17),

wherein the predefined values are further characterized as comprising an address window, each window comprising an initial address and an end address, a match comprising the global address falling between the initial address and the end address (page 6, line 31 to page 7, line 5; Fig. 3, reference characters 201' and 302').

Claim 29. A parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the local data memories of the at least first and second processor elements not on a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor

elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means (page 6, line 5) responsive to writing (page 6, lines 3-5), by the processor of the processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating (page 6, lines 5-9), on the message-passing communications network, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means (page 6, lines 8-9) responsive to receiving a message, via the message-passing communications network, by the communications manager, for comparing (page 6, lines 9-14) the global address in the message with the predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17),

wherein the first means propagates the message to a number of processor elements, the number comprising less than all and more than one of the processor elements (Fig. 3, reference character 311 intersecting less than all and more than one of the processing elements).

Claim 33. A method of operating a parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications

manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0) and not solely by a common bus; the processor elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:

writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element (page 6, lines 3-5);

adding, by the communications manager, a global address to the result of the computation (page 6, line 5);

propagating, on the message-passing communications network and not solely by a common bus, a message comprising the global address and the result of the computation (page 6, lines 5-9);

receiving the message, via the message-passing communications network and not solely by a common bus, by the communications manager of the second processor element (page 6, lines 8-9);

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match (page 6, lines 9-14);

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17).

Claim 34. A parallel computer system having at least first and second processor elements (page 5, line 19, reference characters 1' and 1"), each processor element comprising a processor (page 4, line 14; Fig. 2, reference character 2'), a local program memory (abstract), a local data memory (abstract), a communications manager (page 5, line 24; Fig. 2, reference character 6') and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus (page 5, line 7; Fig. 2, reference character 5'); the local data memories of the at least first and second processor elements not on a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network (page 6, lines 7-8; Fig. 2, reference character 0); the processor elements each executing an application (page 5, line 29); each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means (page 6, line 5) responsive to writing (page 6, lines 3-5), by the processor of the processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating (page 6, lines 5-9), on the message-passing communications network and not solely by a common bus, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means (page 6, lines 8-9) responsive to receiving a message, via the message-passing

communications network and not solely by a common bus, by the communications manager, for comparing (page 6, lines 9-14) the global address in the message with the plurality of predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory (page 6, lines 14-17).

Grounds of rejection to be reviewed on appeal

The final rejection is dated March 24, 2004. In this Office Action the Examiner rejects all of claims 16-34 as supposedly "clearly anticipated" by U.S. Pat. No. 5,117,350 to Parrish et al. ("Parrish").

Argument

This application has had a long prosecution history. A first patent application was filed July 21, 1997 (appl. no. 60/053,207). A second patent application was filed July 23, 1998 (appl. no. 09/121,493). The Examiner granted the courtesy of a first interview on June 13, 2000, at which time the undersigned was under the impression that an allowance would follow shortly. This second patent application was finally rejected, however, and a third application was filed September 12, 2001 (appl. no. 09/954,596) as a continuation thereof. The Examiner granted the courtesy of a second interview on November 22, 2002, at which time the undersigned was again under the impression that an allowance would follow shortly. This third application was finally rejected, however. The Examiner granted the courtesy of a third interview on March 6, 2003, and yet again the undersigned was under the impression that an allowance would follow shortly. The finality of rejection was withdrawn, but was again made final.

It is noted that the Parrish reference, relied upon as supposedly anticipating all pending claims, was first made of record *by the applicant* in an Information Disclosure Statement dated February 1, 2000. This reference was not cited against the pending claims in the

first Office Action (June 6, 2000), the second Office Action (March 13, 2001), the third Office Action (July 3, 2002), or the fourth Office Action (February 13, 2003). Parrish was first cited against the pending claims in a fifth Office Action dated March 28, 2003. The undersigned had by this time traveled to the Patent Office three times to interview this case, with Parrish not mentioned during any of the three interviews. The rejection over Parrish was made final in an Office Action dated March 24, 2004. This appeal, for an invention made over seven years ago, follows.

As Parrish is the sole reference upon which the Examiner relies in rejecting all claims as supposedly anticipated, it is helpful to discuss Parrish in some detail.

It is true that there are some similarities between Parrish and the claimed invention. It is crucially important, however, to appreciate that there are important differences. The most important distinction is that in the claimed invention, there is an express limitation that the processor elements are coupled by means of a message-passing communications network. In contrast, the processor elements of Parris are coupled by means of a bus.

It is noted that both the terms "bus" and "message-passing communications network" are used in the claims as filed and were used in the application as filed. If the two terms were synonymous (which is what would need to be the case for Parrish to anticipate the claims) then there would have been no need for applicant to use such distinct terms in the application as filed. In the application, the inventor included a bus as a possible topology of the network, see specification page 4, line 10.

By way of background it is mentioned that as is well known to those skilled in the art, a message-passing network is more complex and is different from a simple bus. For example, scalability is greater for a message-passing network, and sub-groups and sub-networks can be established. In a bus, mutual influences can happen such as interference and contention for the medium.

The Parrish reference teaches a bus oriented system. The bus structure of the Parrish references allows message passing processes. However, in Parrish, all processor elements (which the Examiner equates to Parrish's "functional units") are on a common bus. Parrish, therefore, does not teach a message-passing communication network. A one-dimensional structure such as a bus is not a network according to a usual definition.

In a bus-oriented system, only one message may be transmitted at one time. In a communications network, hundreds or thousands of messages may be in transit at the same time. The claimed method is therefore much more complex than a bus-oriented system such as that of Parrish.

It is helpful to enumerate the places in the pending claims where the limitation (of the coupling through the message-passing communications network) may be found, a limitation not found in Parrish. These are as follows:

- in claims 16, 26, 27 and 33 at line 7, and again in the "propagating" and "receiving" steps;

- in claims 21, 28, 29 and 34 at line 7, and again in the first means and in the second means.

The Examiner is respectfully requested to note that each pending independent claim (16, 21, 26, 27, 28, 29, 33 and 34) there are at least two express limitations about coupling through a message-passing communications network, and that neither of these limitations may be found in Parrish.

It should be noted that claims 33 and 34 contain wording (three places in each claim) that even more strongly distinguishes over Parrish, reciting that the communications between processor elements is "not solely by a common bus."

For convenience of reference this table is provided, identifying particular limitations to be found in the eight independent claims. Independent claims 16, 26, 27, and 33 are method claims. Independent claims 21, 28, 29, and 34 are apparatus claims. Independent claims 26 and 28 are each limited with respect to an "address window," discussed in more detail below. Independent claims 27 and 29 are each limited with respect to sending messages to "less than all and more than one" of processor elements, again discussed in more detail below. Finally, independent claims 33 and 34 are each limited in that certain communications are made "not solely by a common bus," to be discussed in more detail below.

	method claim	apparatus claim
first claim	16	21
address window	26	28
less than all and more than one	27	29
not solely by a common bus	33	34

The single ground of rejection (supposed anticipation by Parrish) will now be discussed with respect to each of the pending claims in turn.

Argument for claim 16[2]

Claim 16 is:

> A method of operating a parallel computer system having at least first and second processor elements, each processor element comprising a processor, a local program memory, a local data memory, a communications manager and an

2 The Examiner has asked, pursuant to 37 CFR section 41.37(vii), that the subheading for each claim be moved so that it is above the argument rather than in the same line with the argument. For that reason, such subheadings have been moved for this claim argument and for each subsequent claim argument.

operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network; the processor elements each executing an application; each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:

writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element;

adding, by the communications manager, a global address to the result of the computation;

propagating, on the message-passing communications network, a message comprising the global address and the result of the computation;

receiving the message, via the message-passing communications network, by the communications manager of the second processor element;

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match;

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory.

The Examiner has expressed the view that this claim 16 is "clearly anticipated" by Parrish. But for the reasons discussed above (a message-passing communications network is not a bus) it is requested that this rejection be reversed. The rejection of claims 17, 18, 19, 21, 22, 23, 24, 26, 28, 30 and 31 should be reversed for the same reasons as given above with respect to claim 16.

Argument for claim 20

Claim 20 is:

> The method of claim 16 wherein the propagating step comprises propagating the message to a number of processor elements, the number comprising less than all and more than one of the processor elements.

The undersigned has been unable to find this limitation in Parrish. Reversal of this rejection of claim 20 is requested for this reason, and also for the reasons given above in connection with claim 16.

The Examiner suggests (Office Action dated March 28, 2003, page 14) that this limitation is found in Parrish at Fig. 8B-8D, with "a local address of DCM2 allocated in two different nodes." It appears to the undersigned, however, that the addresses to which the Examiner refers (768K to 1280K in Fig. 8B, and 1024K to 1536K in Fig. 8D) cannot possibly represent global addresses. It appears to the undersigned that these are local addresses. As such, an overlap in the start and end points of each locally defined range would not mean anything about global addressing.

Other language in Parrish seems consistent with the view of the undersigned:

At col. 7, lines 38-39, Parrish teaches that "any shared Global Memory partition is housed

on *one and only one* node." This suggests that a write to a Global Memory partition would be a write to only one node, and not to more than one node.

At col. 9, line 5, Parrish talks of a node "reserving" a system physical address, which appears to the undersigned to represent a mechanism that is intended to prevent two or more nodes from "reserving" the same system physical address.

At col. 10, lines 28-33, Parrish says:

> The Remote Global Memory 180 is structured to respond to a contiguous block of system physical addresses with a selected starting and ending address for the block range. When multiple Remote Global Memory modules or units are installed, they each would be assigned a different block address range.

This suggests to the undersigned that Parrish teaching avoiding any two modules being assigned the same block address range. If so, then no single write operation would write to two or more modules at once.

For all these reasons it is suggested that the Examiner has not shown this limitation in Parrish.

Argument for claim 25

Claim 25 is:

> The apparatus of claim 21 wherein the first means propagates the message to a number of processor elements, the number comprising less than all and more than one of the processor elements.

The rejection of claim 25 should be reversed for the same reasons as given above in

connection with claim 16. As discussed above with respect to claim 20, the office actions do not point out where in Parrish this limitation of claim 25 ("less than all and more than one") may be found, and the undersigned has been unable to find this limitation in Parrish. Reversal of this rejection of claim 25 is requested for this additional reason.

Argument for claim 27

Claim 27 is:

> A method of operating a parallel computer system having at least first and second processor elements, each processor element comprising a processor, a local program memory, a local data memory, a communications manager and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network; the processor elements each executing an application; each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:
>
> writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element;
>
> adding, by the communications manager, a global address to the result of the computation;

propagating, on the message-passing communications network, a message comprising the global address and the result of the computation;

receiving the message, via the message-passing communications network, by the communications manager of the second processor element;

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match;

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory,

wherein the propagating step comprises propagating the message to a number of processor elements, the number comprising less than all and more than one of the processor elements.

The rejection of claim 27 should be reversed for the same reason as given above for claim 16, namely the "bus/message-passing-network" distinction. As discussed above with respect to claim 20, the office actions do not point out where in Parrish this limitation of claim 27 ("less than all and more than one") may be found, and the undersigned has been unable to find this limitation in Parrish. Reversal of this rejection of claim 27 is requested for this additional reason.

Argument for claim 29

Claim 29 is:

A parallel computer system having at least first and second processor elements, each processor element comprising a processor, a local program memory, a local

data memory, a communications manager and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus; the local data memories of the at least first and second processor elements not on a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network; the processor elements each executing an application; each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means responsive to writing, by the processor of the processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating, on the message-passing communications network, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means responsive to receiving a message, via the message-passing communications network, by the communications manager, for comparing the global address in the message with the predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory,

wherein the first means propagates the message to a number of processor elements, the number comprising less than all and more than one of the processor elements.

The rejection of claim 29 should be reversed for the same reason as given above for claim 16, namely the "bus/message-passing-network" distinction. As discussed above with respect to claim 20, it appears to the undersigned that the office actions do not point out where in Parrish this limitation of claim 29 ("less than all and more than one") may be found, and the undersigned has been unable to find this limitation in Parrish. Reversal of this rejection of claim 29 is requested for this additional reason.

Argument for claim 33

Claim 33 is:

> A method of operating a parallel computer system having at least first and second processor elements, each processor element comprising a processor, a local program memory, a local data memory, a communications manager and an operating system, within each processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network **and not solely by a common bus**; the processor elements each executing an application; each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory; the method comprising the steps of:
>
> writing, by the processor of the first processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element;
>
> adding, by the communications manager, a global address to the result of the

computation;

propagating, on the message-passing communications network **and not solely by a common bus**, a message comprising the global address and the result of the computation;

receiving the message, via the message-passing communications network **and not solely by a common bus**, by the communications manager of the second processor element;

comparing, by the communications manager of the second processor element, the global address in the message with the predefined values for a match;

in the event of a match, computing a local address by the communications manager of the second processor element, and storing the results of the computation at the local address via the common bus to the local data memory.

The rejection of claim 33 should be reversed for the same reason as given above for claim 16, namely the "bus/message-passing-network" distinction. But it should be noted that this claim differs from claim 16 in expressly reciting in three places (shown in bold type) the fact of the message-passing communications network being quite different from a common bus.

Argument for claim 34

Claim 34 is:

A parallel computer system having at least first and second processor elements, each processor element comprising a processor, a local program memory, a local data memory, a communications manager and an operating system, within each

processor element the local program memory, local data memory, and communications manager all communicatively coupled by means of a common bus; **the local data memories of the at least first and second processor elements not on a common bus**; the communications managers of the at least first and second processor elements communicatively coupled by means of a message-passing communications network; the processor elements each executing an application; each communications manager further containing a plurality of predefined values each of which, in the event of a match of one of the predetermined values to a global address in a message, causes storage of results of an associated computation in local data memory;

for each communications manager, the communications manager comprising first means responsive to writing, by the processor of the processor element, by means of the common bus of the first processor element, a result of a computation into the communications manager of the first processor element, by adding a global address to the result of the computation, and by propagating, on the message-passing communications network **and not solely by a common bus**, a message comprising the global address and the result of the computation;

for each communications manager, the communications manager comprising second means responsive to receiving a message, via the message-passing communications network **and not solely by a common bus,** by the communications manager, for comparing the global address in the message with the plurality of predefined values for a match, in the event of a match, for computing a local address, and storing the results of the computation at the local address via the common bus to the local data memory.

The rejection of claim 34 should be reversed for the same reason as given above for claim 16, namely the "bus/message-passing-network" distinction. But it should be noted that this claim differs from claim 16 in expressly reciting in three places (shown in bold type)

the fact of the message-passing communications network being quite different from a common bus.

So far as independent claims 33 and 34 are concerned, if the rejection over Parrish were to be affirmed, it would be necessary to show where Parrish teaches a system in which the messages are passed between processor elements that are not coupled by means of a common bus. The office actions doe not show where such a teaching is found, and the undersigned is not able to find it in the cited reference. These limitations suffice to distinguish fully over Parrish which teaches specifically that all functional units are on a common bus.

Respectfully submitted,

Carl Oppedahl
PTO Reg. No. 32,746
Oppedahl & Larson LLP
P O Box 5068
Dillon, CO 80435-5068
email oppedahl@patents.com